

1993

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

OCTAVE: A MARSYAS POST-PROCESSOR FOR
COMPUTER-AIDED CONTROL SYSTEM DESIGN

Prepared by:	A. Scottedward Hodel, Ph. D.
Academic Rank:	Assistant Professor
Institution and	Department of Electrical Engineering
Department	Auburn University
MSFC Colleague(s):	D. P. Vallely
NASA/MSFC:	
Office:	Structures and Dynamics Laboratory
Division:	Control System Division
Branch:	Mechanical Systems Control Branch

1 Introduction

MARSYAS is a computer-aided control system analysis package for the simulation and analysis of dynamic systems. In the summer of 1991 MARSYAS was updated to allow for the analysis of sampled-data systems in terms of frequency response, stability, etc. This update was continued during the summer of 1992 in order to extend further MARSYAS commands to the study of sampled-data systems. Further work was done to examine the computation of openat transfer functions, root-locii and w -plane frequency response plots. At the conclusion of the summer 1992 work it was proposed that control-system design capability be incorporated into the MARSYAS package. It was decided at that time to develop a separate "stand-alone" computer-aided control system design (CACSD) package. This report is a brief description of such a package.

A popular CACSD design environment is provided with commercial versions of Matlab, e.g., Simulink (tm) by the MathWorks. The Matlab design environment comprises (1) a compiled main program with a command parser and necessary intrinsic functions for matrix data manipulation, and (2) command scripts, called m-files, which may be used in a fashion similar to Unix shell scripts in order to create an increased function set for the user. The MathWorks has developed several "toolboxes," or sets of such m-files, for specific purposes such as signal processing, state-space control system design, robust control, etc. Since m-files are text-file scripts, their source code is available for viewing by the user. However, source code for any commercial Matlab is proprietary to the vendor and is not available.

In 1992, John Eaton, a post-doc at the University of Texas, began development of a free-ware Matlab look-alike program to be made available under the same licensing terms as that of the Free Software Foundation. That is, the program cannot be sold in whole or in part, and its source code must be freely made available. The numerical routines in Octave are taken from accepted FORTRAN routines in packages such as EISPACK, LINPACK, LAPACK, and the user interface and command execution routines are written in C++ and C. Under a follow-on grant from MSFC, work was begun at Auburn University on preliminary versions of Octave to incorporate new functions into Octave that would aid in the development of a control systems toolbox for this program. This work was continued during the Summer Faculty Fellowship Program during summer of 1993; all code developed was submitted and incorporated into the official Octave distribution. The code development is still ongoing; however, the design environment provided by the current version (0.74.5) is sufficiently functional that it can be used for a wide variety of applications. Version 1.0 of Octave is expected to be released shortly (prior to the end of 1993).

The remainder of this report is organized as follows. In Section 2 is presented a description of the planned MARSYAS design environment. Following this, in Section 3 a design example using current MARSYA/OCTAVE functions is presented. Finally, in Section 4 we discuss planned enhancements to the MARSYAS/OCTAVE system.

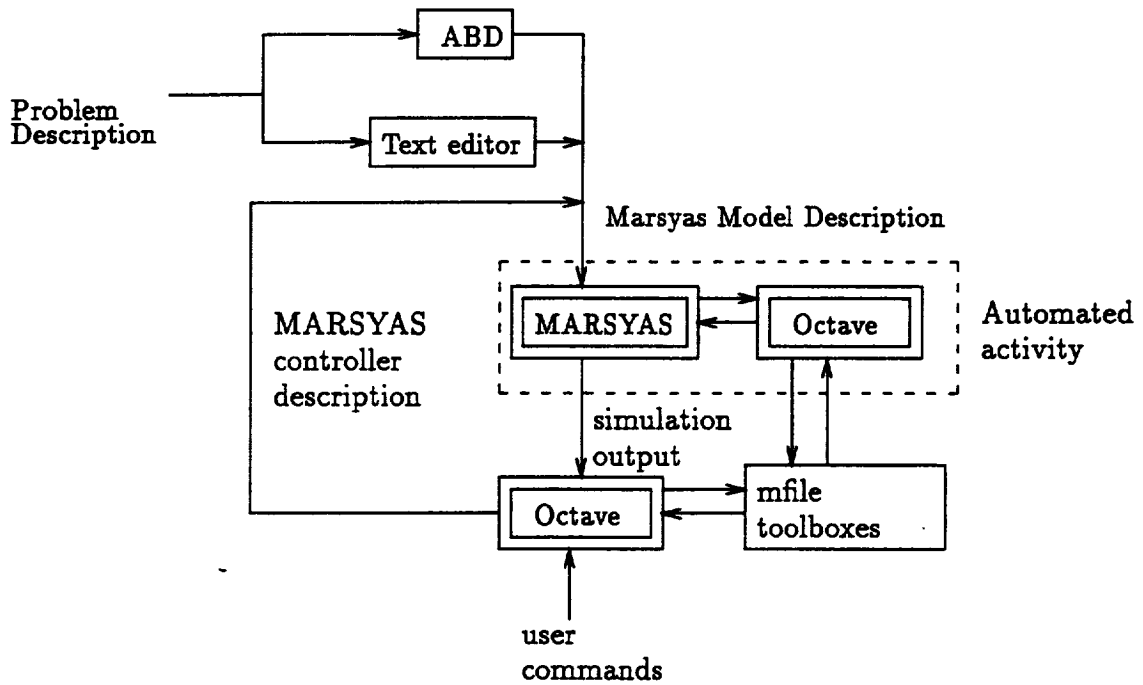


Figure 1: Desired MARSYAS design environment

2 Planned MARSYAS Design Environment

The desired MARSYAS design environment is shown in Figure 1. Those portions that are under development are shown in dashed-lines, those that are planned are shown in dotted lines. The user, once determining their problem description, writes a MARSYAS model description of the corresponding dynamic system. MARSYAS is run as a batch process; while not currently implemented, it is planned to modify MARSYAS in order to allow the Marsyas analysis phase to make use of Octave. The results of the analysis and simulation phase of MARSYAS are read into OCTAVE via m-file `marsyas_in.m`, which currently loads the system linearization (A, B, C, D) for either continuous or discrete-time systems. From within Octave, the user interactively uses m-file scripts in order to design a controller that meets desired design criteria, and then uses the m-file `marsyas_out` to store a MARSYAS model description of the designed controller. This controller may then be verified with the nonlinear MARSYAS model description with a subsequent MARSYAS run, and further controller modifications may be made interactively from within OCTAVE.

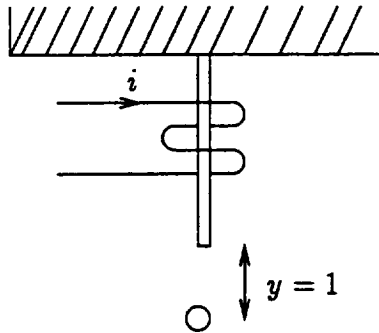


Figure 2: Magnetically suspended ball

3 Design example

The Octave design toolbox currently contains only one function: linear quadratic Gaussian (LQG) controller design. As an example of the MARSYAS/Octave design environment, consider the magnetically suspended ball system shown in Figure 2. The corresponding MARSYAS description module is

```

CONSTANT: G = 9.8$
MODEL: BALL\DYNAMICS,EQUATION$
INPUTS: IM $
OUTPUTS: X,XDOT$
EQUATION: X'' = G - (IM**2)/(X**2)$
          : XDOT = X' $
END$

```

A MARSYAS simulation was run to obtain a linearization of the above non-linear system, and the resulting data were employed by the following Octave m-file:

```

[a,b,c,d] = marsyas_in()
[n,m] = size(b);
[p,m] = size(d);
disp('open loop poles:')
poles= eig(a)'
% state feedback design
[k,x,e] = lqr(a,b,eye(n),10*eye(m));
disp('closed-loop state-feedback poles are')
poles = eig(a-b*k)'
% state estimator design
[l,x2,e] = lqe(a,eye(n),c,eye(n),0.01*eye(p))

```

```

bc = l';
cc = k;
dc = zeros(m,p);
ac = a - l'*c - b*cc;
marsyas_out(ac,bc,cc,dc)

```

The commands `m`, `arsyas_in` and `marsyas_out` are used to interact with the MARSYAS program, and the Octave m-files `lqr` and `lqe` are Octave scripts that solve the appropriate algebraic Riccati equations in order to obtain the desired controller. The MARSYAS controller description thus obtained is

```

MODEL: OCTAVE, EQUATION $
INPUTS: U1, U2$
* 1: X $
* 2: XDOT $
OUTPUTS: Y1$
* 1: I\MAG $
EQUATION: X1' = -2.700417E+01 * X1 -2.653148E+01 * X2
          + 6.831738E+00 * U1 + 1.792014E+01 * U2 $
          : X2' = -5.831738E+00 * X1 -8.184793E+00 * X2
          + 8.184793E+00 * U1 + 6.831738E+00 * U2 $
          : Y1 = -1.450893E+00 * X1 -6.276922E+00 * X2 $
END$

```

and is incorporated into the original simulation by adding a main model block:

```

MODEL: MAIN, EQUATION$
INPUTS: I\MAG $
OUTPUTS: X,XDOT$
EQUATION: IM = I\MAG - ID $
          : XERR = X-1 $
SUBMODEL: BALL\DYNAMICS; INPUTS: IM; OUTPUTS: X,XDOT $
SUBMODEL: OCTAVE; INPUTS: XERR,XDOT; OUTPUTS: ID $
END$

```

4 Planned Work

Planned enhancements to the MARSYAS Octave environment include

1. advanced design options,
2. improved user documentation (on-line and off-line), and
3. absorption of MARSYAS analysis phase into Octave

Ultimately, it is expected that Octave will prove itself as a good production code for use in control system design at MSFC.